Data Cleaning Course Project

Duncan Turnbull

20 December 2014

Overview

The project requirement is to take raw data and turn it into tidy data. The data is an online zip file and is part of the "Human Activity Recognition Using Smartphones Data Set". It has come from a group of 30 volunteers aged between 19 & 48. The data is the accelerations recorded as they move using a smartphone accelerometer and gyroscope.

Project Requirements

You should create one R script called run_analysis.R that does the following.

1. Merges the training and the test sets to create one data set.
2. Extracts only the measurements on the mean and standard deviation for each measurement.
3. Uses descriptive activity names to name the activities in the data set
4. Appropriately labels the data set with descriptive variable names.
5. From the data set in step 4, creates a second, independent tidy data set with the average of each variable for each activity and each subject.

Part 1

- Download data and unzip
- load test and training sets
  - load row and column labels
    - get subset of labels that is mean and std before cleaning
    - clean column and row labels
    - apply labels
  - add variable to reflect whether test or train data
- Merge data

Part 2: select std deviation and mean data only

- Once all data is in one table discard non necessary columns or
- could try reading in only required columns or remove straight away
  - would require an early logical vector of which columns are required.
  - whats the best way to do this?
  - looking for mean() and std() - could get caught by meanfreq() if dont use ()

Part 3: Name the activities appropriately

- Need to define all included activities - walk, run, upstairs etc, these are the rows
- Add column called ActivityName which links from activity.txt

Part 4

- Label variable names (columns)

Part 5

- create a second, independent tidy data set with the average of each variable for each activity and each subject.

Overview

There are a range of files that need to be merged. The observations are raw data, there is a user file linking the user ids to the rows

The file names reflect the data types and the directories they are in also split between the test and train group

Part 1

0. Setup parameters
    - Setup directories to access data folder
    - Assumptions:
        - we are starting in the working dir
        - we know specific files names and mapping to directories
1. Read labels and clean them
    - Move to "UCI HAR Dataset" dir. This is the base directory
    - read the files "activity_labels.txt" ,"features.txt".
        - these are needed to label rows and set column variable name
    - Read the label files (and clean them if needed)
    - Read the files into data tables and add vars for test/train (although it doesn't look like we need this anywhere)
    - apply the label from the files to the data
2. Read files

```
activitylabels <- read.table(file = "activity_labels.txt")
# Get the required activity columns
features <- read.table(file = "./features.txt")
wantedcols <- grep("(mean|std)\\(\\)",features$V2)
featurelabels <- gsub(features[,2],pattern = "[-(),]", replacement = ""  )

ytrain <- read.table(file = "./train//y_train.txt")
ytest <- read.table(file = "./test/y_test.txt")
activity <- rbind(ytest,ytrain)

activitynames <- as.character(activitylabels$V2)[activity$V1]

testsubject <- read.table(file = "./test//subject_test.txt")
trainsubject <- read.table(file = ".//train//subject_train.txt")
subject <- rbind(testsubject, trainsubject)
```

```
# Get the big test set of data
xtest <- read.table(file = "./test/X_test.txt")
# Set the names for the test data
colnames(xtest) <- features$V2
# Get the big train set of data
xtrain <- read.table(file = "./train/X_train.txt")
# Set the names for the train data
colnames(xtrain) <- features$V2
# Repeat the type variable type to show this is train data

# Add a variable type to show whether test or train data
type <- character(nrow(xtest)+nrow(xtrain))
type[1:nrow(xtest)] <- "Test"
type[nrow(xtest)+1:nrow(xtrain)] <- "Train"

# Create the big set of data
data <- rbind(xtest,xtrain)
# Capture the current names of the data
names <- names(data)
# Join the data with the subject, activity and activity name ( for easier reading)
data <- cbind(subject,activity,activitynames,type,data)
# Update the names
colnames(data) <-c("Subject","Activity","ActivityName","Type", as.character(names))

# Use previously selected active columns to remove anything but mean or std
data2 <-data[,c(1:4,actcols+4)]

# Clean the variable names
names2 <-names(data2)
names2 <- gsub(names2,pattern = "[-(),]", replacement = ""  )
names(data2) <- names2
```

This gets us to Part 4 (merged 3 parts together back there)

Now we need to create a second independent data set with average of each variable for each activity and each subject:

- This seems like a good call to use dplyr group_by function to group by activity and subject and summarise to process everything
- What is needed is to group by activity and within that subject then create the average for each variable (column)
- The new data will have 6 activities X 30 subject = 180 rows and 66 averages per row

```
library(dplyr)

tidy <- group_by(data2, ActivityName, Subject)
tidyoutput <-summarise_each(tbl = tidy, funs(mean), c(-Type,-Activity))

# Write out the tidy data
write.table( tidyoutput, row.name=FALSE ,file = "./tidyoutput.txt")
```

Howto reading the output file back in

To read the file back in use:

```
View( read.table(file="tidyoutput.txt", header=T) )
```

Tidy data

This output is tidy.

- One observation per row
- One variable per column
- The observational units form a table

Other clean up included: * Cleaned variable names of non alpha characters “(),_" * Only selection of required columns

References:

- https://class.coursera.org/getdata-016/forum/thread?thread_id=100
- http://vita.had.co.nz/papers/tidy-data.pdf